

NSD PROJECT2 DAY01

1. [案例1：配置逻辑卷](#)
2. [案例2：配置数据库服务器](#)
3. [案例3：配置主从同步](#)
4. [案例4：配置读写分离服务](#)
5. [案例5：准备NFS服务存储磁盘](#)
6. [案例6：配置NFS服务](#)
7. [案例7：配置网站服务](#)
8. [案例8：测试配置](#)
9. [案例9：配置监控服务](#)
10. [案例10：具体配置如下\(以网站服务器192.168.4.33为例\)：](#)
11. [案例11：监控配置](#)

1 案例1：配置逻辑卷

1.1 问题

具体配置如下：

- 添加磁盘
- 磁盘分区
- 创建LV
- 格式化

1.2 方案

分别给2台虚拟机添加2块5G的磁盘。如图-1所示。

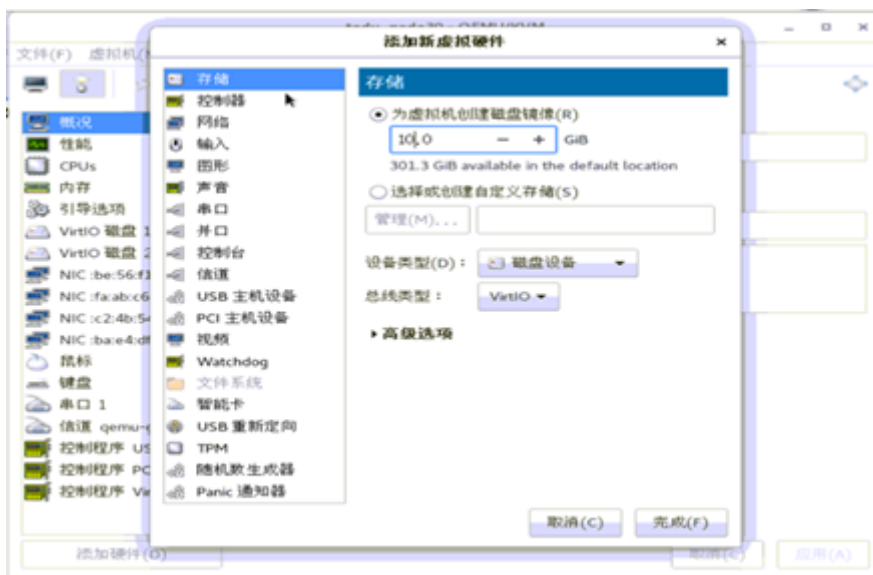


图-1

1.3

1.4 步骤

实现此案例需要按照如下步骤进行。

[Top](#)

步骤一：磁盘分区(2台数据库服务器都配置)

1) 创建分区，每块盘分1个区即可

```
01. [root@mysql11 ~]# fdisk -l /dev/vdb //查看磁盘容量
02.
03. 磁盘 /dev/vdb: 5368 MB, 5368709120 字节, 10485760 个扇区
04. Units = 扇区 of 1 * 512 = 512 bytes
05. 扇区大小(逻辑/物理): 512 字节 / 512 字节
06. I/O 大小(最小/最佳): 512 字节 / 512 字节
07.
08. [root@mysql11 ~]#
09. [root@mysql11 ~]# fdisk -l /dev/vdc //查看磁盘容量
10.
11. 磁盘 /dev/vdc: 5368 MB, 5368709120 字节, 10485760 个扇区
12. Units = 扇区 of 1 * 512 = 512 bytes
13. 扇区大小(逻辑/物理): 512 字节 / 512 字节
14. I/O 大小(最小/最佳): 512 字节 / 512 字节
15.
16. [root@mysql11 ~]#
17.
18. [root@mysql11 ~]# fdisk /dev/vdb //磁盘分区
19. 欢迎使用 fdisk (util-linux 2.23.2)。
20.
21. 更改将停留在内存中，直到您决定将更改写入磁盘。
22. 使用写入命令前请三思。
23.
24. Device does not contain a recognized partition table
25. 使用磁盘标识符 0x882d04b7 创建新的 DOS 磁盘标签。
26.
27. 命令(输入 m 获取帮助): n //创建分区
28. Partition type:
29.   p   primary (0 primary, 0 extended, 4 free)
30.   e   extended
31. Select (default p): p //创建主分区
32. 分区号 (1-4, 默认 1): 1 //指定编号
33. 起始 扇区 (2048-10485759, 默认为 2048): //起始柱面数
34. 将使用默认值 2048
35. Last 扇区, +扇区 or +size {K,M,G} (2048-10485759, 默认为 10485759): //结束柱
36. 将使用默认值 10485759 Top
37. 分区 1 已设置为 Linux 类型，大小设为 5 GiB
```

```
38.
39. 命令(输入 m 获取帮助): w //保存退出
40. The partition table has been altered!
41.
42. Calling ioctl() to re-read partition table.
43. 正在同步磁盘。
44. [root@mysql11 ~]#
45.
46.
47. [root@mysql11 ~]# fdisk /dev/vdc //给vdc盘分区
48. 欢迎使用 fdisk (util-linux 2.23.2)。
49.
50. 更改将停留在内存中，直到您决定将更改写入磁盘。
51. 使用写入命令前请三思。
52.
53. Device does not contain a recognized partition table
54. 使用磁盘标识符 0x5184f504 创建新的 DOS 磁盘标签。
55.
56. 命令(输入 m 获取帮助): n
57. Partition type:
58.   p   primary (0 primary, 0 extended, 4 free)
59.   e   extended
60. Select (default p): p
61. 分区号 (1-4, 默认 1): 1
62. 起始 扇区 (2048-10485759, 默认为 2048):
63. 将使用默认值 2048
64. Last 扇区, +扇区 or +size {K,M,G} (2048-10485759, 默认为 10485759):
65. 将使用默认值 10485759
66. 分区 1 已设置为 Linux 类型，大小设为 5 GiB
67.
68. 命令(输入 m 获取帮助): w
69. The partition table has been altered!
70.
71. Calling ioctl() to re-read partition table.
72. 正在同步磁盘。
73. [root@mysql11 ~]#
```

2) 查看分区

[Top](#)

```
01. [root@mysql11 ~]# fdisk -l /dev/vdb
```

```

02.
03. 磁盘 /dev/vdb: 5368 MB, 5368709120 字节, 10485760 个扇区
04. Units = 扇区 of 1 * 512 = 512 bytes
05. 扇区大小(逻辑/物理): 512 字节 / 512 字节
06. I/O 大小(最小/最佳): 512 字节 / 512 字节
07. 磁盘标签类型: dos
08. 磁盘标识符: 0x882d04b7
09.
10.      设备 Boot      Start          End      Blocks   Id  System
11.  /dev/vdb1          2048      10485759      5241856    83  Linux
12. [root@mysql11 ~]#
13. [root@mysql11 ~]#
14. [root@mysql11 ~]# fdisk -l /dev/vdc
15.
16. 磁盘 /dev/vdc: 5368 MB, 5368709120 字节, 10485760 个扇区
17. Units = 扇区 of 1 * 512 = 512 bytes
18. 扇区大小(逻辑/物理): 512 字节 / 512 字节
19. I/O 大小(最小/最佳): 512 字节 / 512 字节
20. 磁盘标签类型: dos
21. 磁盘标识符: 0x5184f504
22.
23.      设备 Boot      Start          End      Blocks   Id  System
24.  /dev/vdc1          2048      10485759      5241856    83  Linux
25. [root@mysql11 ~]#

```

步骤二：创建LV

1) 修改设备ID

```

01. [root@mysql11 ~]# fdisk /dev/vdb //修改/dev/vdb
02. 欢迎使用 fdisk (util-linux 2.23.2)。
03.
04. 更改将停留在内存中，直到您决定将更改写入磁盘。
05. 使用写入命令前请三思。
06.
07.
08. 命令(输入 m 获取帮助): t //修改类型
09. 已选择分区 1
10. Hex 代码(输入 L 列出所有代码): 8e //指定为8e
11. 已将分区“Linux”的类型更改为“Linux LVM”
12. 命令(输入 m 获取帮助): w //保存退出

```

[Top](#)

```
13. The partition table has been altered!
14.
15. Calling ioctl() to re-read partition table.
16. 正在同步磁盘。
17. [root@mysql11 ~]# fdisk -l /dev/vdb //查看磁盘信息
18.
19. 磁盘 /dev/vdb: 5368 MB, 5368709120 字节, 10485760 个扇区
20. Units = 扇区 of 1 * 512 = 512 bytes
21. 扇区大小(逻辑/物理): 512 字节 / 512 字节
22. I/O 大小(最小/最佳): 512 字节 / 512 字节
23. 磁盘标签类型: dos
24. 磁盘标识符: 0x882d04b7
25.
26.      设备 Boot      Start          End      Blocks   Id  System
27. /dev/vdb1           2048      10485759      5241856    8e  Linux LVM
28. [root@mysql11 ~]#
29.
30.
31.
32. [root@mysql11 ~]# fdisk /dev/vdc
33. 欢迎使用 fdisk (util-linux 2.23.2)。
34.
35. 更改将停留在内存中，直到您决定将更改写入磁盘。
36. 使用写入命令前请三思。
37.
38.
39. 命令(输入 m 获取帮助): t
40. 已选择分区 1
41. Hex 代码(输入 L 列出所有代码): 8e
42. 已将分区“Linux”的类型更改为“Linux LVM”
43.
44. 命令(输入 m 获取帮助): w
45. The partition table has been altered!
46.
47. Calling ioctl() to re-read partition table.
48. 正在同步磁盘。
49. [root@mysql11 ~]# fdisk -l /dev/vdc
50.
51. 磁盘 /dev/vdc: 5368 MB, 5368709120 字节, 10485760 个扇区
52. Units = 扇区 of 1 * 512 = 512 bytes
53. 扇区大小(逻辑/物理): 512 字节 / 512 字节
```

[Top](#)

```

54. I/O 大小(最小/最佳): 512 字节 / 512 字节
55. 磁盘标签类型: dos
56. 磁盘标识符: 0x5184f504
57.
58.      设备 Boot      Start      End      Blocks  Id  System
59.  /dev/vdc1      2048     10485759     5241856   8e  Linux LVM
60. [root@mysql11 ~]#

```

3) 安装软件提供命令的软件

```

01.
02. [root@mysql11 ~]# yum -y install lvm2 //安装软件
03. 已加载插件: fastestmirror
04. Loading mirror speeds from cached hostfile
05. local_repo | 3.
06. 正在解决依赖关系
07. --> 正在检查事务
08. ---> 软件包 lvm2.x86_64.7.2.02.177-4.el7 将被 安装
09. .....
10. .....
11. 已安装:
12.     lvm2.x86_64 7:2.02.177-4.el7
13.
14. 作为依赖被安装:
15.     device-mapper-event.x86_64 7:1.02.146-4.el7
16.     device-mapper-event-libs.x86_64 7:1.02.146-4.el7
17.     device-mapper-persistent-data.x86_64 0:0.7.3-3.el7
18.     libaio.x86_64 0:0.3.109-13.el7
19.     lvm2-libs.x86_64 7:2.02.177-4.el7
20.
21. 完毕!
22. [root@mysql11 ~]#

```

4) 创建PV

```

01. [root@mysql11 ~]#
02. [root@mysql11 ~]# pvcreate /dev/vdb1 /dev/vdc1 //创建PV Top
03. Physical volume "/dev/vdb1" successfully created.
04. Physical volume "/dev/vdc1" successfully created.

```

```

05. [root@mysql11 ~]#
06. [root@mysql11 ~]# pvdisplay //查看PV信息
07.      "/dev/vdb1" is a new physical volume of "<5.00 GiB"
08.      --- NEW Physical volume ---
09.      PV Name                /dev/vdb1
10.      VG Name
11.      PV Size                <5.00 GiB
12.      Allocatable           NO
13.      PE Size                0
14.      Total PE               0
15.      Free PE                0
16.      Allocated PE          0
17.      PV UUID                nDR2To-tDoy-Vow6-AZz9-ryXe-go8D-RJLsdT
18.
19.      "/dev/vdc1" is a new physical volume of "<5.00 GiB"
20.      --- NEW Physical volume ---
21.      PV Name                /dev/vdc1
22.      VG Name
23.      PV Size                <5.00 GiB
24.      Allocatable           NO
25.      PE Size                0
26.      Total PE               0
27.      Free PE                0
28.      Allocated PE          0
29.      PV UUID                OLxKI6-OymP-erzX-2sFh-6R4Z-TU1I-A0Z3C1
30.
31. [root@mysql11 ~]#

```

5) 创建VG

```

01. [root@mysql11 ~]# vgcreate vg0 /dev/vdb1 /dev/vdc1 //创建vg 名称vg0
02.      Volume group "vg0" successfully created
03. [root@mysql11 ~]# vgdisplay
04.      --- Volume group ---
05.      VG Name                vg0
06.      System ID
07.      Format                  lvm2
08.      Metadata Areas          2
09.      Metadata Sequence No    1
10.      VG Access               read/write

```

[Top](#)

```

11.    VG Status          resizable
12.    MAX LV             0
13.    Cur LV             0
14.    Open LV            0
15.    Max PV             0
16.    Cur PV             2
17.    Act PV             2
18.    VG Size            9.99 GiB
19.    PE Size            4.00 MiB
20.    Total PE           2558
21.    Alloc PE / Size    0 / 0
22.    Free PE / Size     2558 / 9.99 GiB
23.    VG UUID            IEduUc-jUgs-pNcs-En2M-XZLc-xpsx-GC3Xhc
24.
25.    [root@mysql11 ~]#

```

6) 创建LV

```

01.    [root@mysql11 ~]# lvcreate -L 9.99G -n lv0 vg0 //创建LV 名称为lv0 大小9.9
02.    Rounding up size to full physical extent 9.99 GiB
03.    Logical volume "lv0" created.
04.    [root@mysql11 ~]# lvdisplay
05.    --- Logical volume ---
06.    LV Path                /dev/vg0/lv0
07.    LV Name                 lv0
08.    VG Name                 vg0
09.    LV UUID                 JTMuKS-b8b0-6dMK-VKqA-lji0-rN59-FPVLch
10.    LV Write Access        read/write
11.    LV Creation host, time mysql11, 2019-07-05 09:39:41 +0800
12.    LV Status               available
13.    # open                  0
14.    LV Size                 9.99 GiB
15.    Current LE              2558
16.    Segments                2
17.    Allocation              inherit
18.    Read ahead sectors      auto
19.    - currently set to      256
20.    Block device            252:0
21.
22.    [root@mysql11 ~]#

```

[Top](#)

7) 格式化

```

01. [root@mysql11 ~]# mkfs.xfs /dev/vg0/lv0 //格式化为 xfs 文件系统
02. meta-data=/dev/vg0/lv0          isize=512    agcount=4, agsize=654848 blks
03.          =                       sectsz=512    attr=2, projid32bit=1
04.          =                       crc=1         finobt=0, sparse=0
05. data          =                   bsize=4096    blocks=2619392, imaxpct=25
06.          =                       sunit=0       swidth=0 blks
07. naming        =version 2          bsize=4096    ascii-ci=0 ftype=1
08. log           =internal log       bsize=4096    blocks=2560, version=2
09.          =                       sectsz=512    sunit=0 blks, lazy-count=1
10. realtime      =none              extsz=4096    blocks=0, rtextents=0
11.
12. [root@mysql11 ~]# blkid /dev/vg0/lv0 //查看文件系统类型
13. /dev/vg0/lv0: UUID="3e06f58f-8793-4800-b444-676f5a07fe10" TYPE="xfs"
14. [root@mysql11 ~]#

```

2 案例2：配置数据库服务器

2.1 问题

具体操作如下：

1. 安装MySQL软件
2. 挂载LV分区
3. 启动服务
4. 管理员登录

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装软件MySQL服务软件(2台数据库服务器都要安装)

1) 解包软件

```

01. [root@mysql11 ~]# tar -xvf mysql-5.7.17.tar
02. ./mysql-community-client-5.7.17-1.el7.x86_64.rpm
03. ./mysql-community-common-5.7.17-1.el7.x86_64.rpm
04. ./mysql-community-devel-5.7.17-1.el7.x86_64.rpm
05. ./mysql-community-embedded-5.7.17-1.el7.x86_64.rpm
06. ./mysql-community-embedded-compat-5.7.17-1.el7.x86_64.rpm
07. ./mysql-community-embedded-devel-5.7.17-1.el7.x86_64.rpm

```

[Top](#)

```
08. ./mysql-community-libs-5.7.17-1.el7.x86_64.rpm
09. ./mysql-community-libs-compat-5.7.17-1.el7.x86_64.rpm
10. ./mysql-community-minimal-debuginfo-5.7.17-1.el7.x86_64.rpm
11. ./mysql-community-server-5.7.17-1.el7.x86_64.rpm
12. ./mysql-community-test-5.7.17-1.el7.x86_64.rpm
13. [root@mysql11 ~]#
14.
15. [root@mysql11 ~]# ls *.rpm
16. mysql-community-client-5.7.17-1.el7.x86_64.rpm
17. mysql-community-common-5.7.17-1.el7.x86_64.rpm
18. mysql-community-devel-5.7.17-1.el7.x86_64.rpm
19. mysql-community-embedded-5.7.17-1.el7.x86_64.rpm
20. mysql-community-embedded-compat-5.7.17-1.el7.x86_64.rpm
21. mysql-community-embedded-devel-5.7.17-1.el7.x86_64.rpm
22. mysql-community-libs-5.7.17-1.el7.x86_64.rpm
23. mysql-community-libs-compat-5.7.17-1.el7.x86_64.rpm
24. mysql-community-minimal-debuginfo-5.7.17-1.el7.x86_64.rpm
25. mysql-community-server-5.7.17-1.el7.x86_64.rpm
26. mysql-community-test-5.7.17-1.el7.x86_64.rpm
27. [root@mysql11 ~]#
```

2) 安装软件

```
01. [root@mysql11 ~]# yum -y install mysql-community-*.rpm
02. 已加载插件：fastestmirror
03. 正在检查 mysql-community-client-5.7.17-1.el7.x86_64.rpm: mysql-community-cl
04. mysql-community-client-5.7.17-1.el7.x86_64.rpm 将被安装
05. .....
06. .....
07. 已安装:
08.     mysql-community-client.x86_64 0:5.7.17-1.el7
09.     mysql-community-devel.x86_64 0:5.7.17-1.el7
10.     mysql-community-embedded-compat.x86_64 0:5.7.17-1.el7
11.     mysql-community-libs.x86_64 0:5.7.17-1.el7
12.     mysql-community-minimal-debuginfo.x86_64 0:5.7.17-1.el7
13.     mysql-community-test.x86_64 0:5.7.17-1.el7
14.
15. 作为依赖被安装:
16.     perl-Data-Dumper.x86_64 0:2.145-3.el7
17.
```

[Top](#)

```
18.    完毕！
19.    [root@mysql11 ~]#
```

步骤二：挂载lv设备

1) 设置开机挂载

```
01.    [root@mysql11 ~]# blkid /dev/vg0/lv0
02.    /dev/vg0/lv0: UUID="3e06f58f-8793-4800-b444-676f5a07fe10" TYPE="xfs"
03.    [root@mysql11 ~]#
04.
05.    [root@mysql11 ~]# vim /etc/fstab
06.    /dev/vg0/lv0    /var/lib/mysql xfs        defaults        0 0
07.    :wq
08.    [root@mysql11 ~]#
```

2) 查看挂载信息

```
01.    [root@mysql11 ~]# mount -a
02.
03.    [root@mysql11 ~]# mount | grep lv0
04.    /dev/mapper/vg0-lv0 on /var/lib/mysql type xfs (rw,relatime,attr2,inode64,no
05.    [root@mysql11 ~]#
```

步骤三：启动服务

1) 启动服务

2台数据库服务器都要启动服务

```
01.    [root@mysql11 ~]# systemctl start mysqld
02.    [root@mysql11 ~]#
03.    [root@mysql11 ~]# ls /var/lib/mysql
04.    auto.cnf    client-cert.pem ibdata1        ibtmp1        mysql.sock.lock  pu
05.    ca-key.pem  client-key.pem  ib_logfile0    mysql         performance_schema se
06.    ca.pem      ib_buffer_pool ib_logfile1    mysql.sock    private_key.pem  se
07.    [root@mysql11 ~]#
08.    [root@mysql11 ~]# systemctl enable mysqld
09.    [root@mysql11 ~]#
10.
```

[Top](#)

```

11. [root@mysql11 ~]# netstat -utnlp | grep :3306
12. tcp6          0          0 :::3306          :::*             LISTEN
13. [root@mysql11 ~]#

```

步骤四：管理员登录

1) 查看初始密码

```

01. [root@mysql11 ~]# grep password /var/log/mysqld.log
02. 2019-07-05T01:56:51.895852Z 1 [Note] A temporary password is generated for
03.
04. [root@mysql11 ~]#
05. [root@mysql11 ~]# mysql -uroot -p'bB0*uCmu:.Kj' //初始密码登录
06. mysql: [Warning] Using a password on the command line interface can be insecure
07. Welcome to the MySQL monitor.  Commands end with ; or \g.
08. Your MySQL connection id is 3
09. Server version: 5.7.17
10.
11. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved
12.
13. Oracle is a registered trademark of Oracle Corporation and/or its
14. affiliates. Other names may be trademarks of their respective
15. owners.
16.
17. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement
18.
19. mysql>

```

2) 修改登录密码

```

01. mysql> alter user root@"localhost" identified by "123qqq...A";
02. Query OK, 0 rows affected (0.01 sec)
03. mysql>
04. mysql> exit
05. Bye
06. [root@mysql11 ~]#

```

[Top](#)

3) 新密码登录

```
01. [root@mysql11 ~]# mysql -uroot -p123qqq...A
02. mysql: [Warning] Using a password on the command line interface can be insecure
03. Welcome to the MySQL monitor.  Commands end with ; or \g.
04. Your MySQL connection id is 4
05. Server version: 5.7.17 MySQL Community Server (GPL)
06.
07. Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved
08.
09. Oracle is a registered trademark of Oracle Corporation and/or its
10. affiliates. Other names may be trademarks of their respective
11. owners.
12.
13. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement
14.
15. mysql> show databases;
16. +-----+
17. | Database          |
18. +-----+
19. | information_schema |
20. | mysql              |
21. | performance_schema |
22. | sys                |
23. +-----+
24. 4 rows in set (0.00 sec)
```

3 案例3：配置主从同步

3.1 问题

配置步骤如下：

1. 配置主服务器
2. 配置从服务器

3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置主服务器

1) 启用binlog日志

```
01. [root@mysql11 ~]# vim /etc/my.cnf
02. [mysqld]
03. server_id=11
```

[Top](#)

```

04. log-bin=master11
05. :wq
06. [root@mysql11 ~]# systemctl restart mysqld
07. [root@mysql11 ~]#

```

2) 用户授权

```

01. [root@mysql11 ~]# mysql -uroot -p123qqq...A
02. mysql: [Warning] Using a password on the command line interface can be insecure
03. Welcome to the MySQL monitor.  Commands end with ; or \g.
04. Your MySQL connection id is 3
05. mysql>
06. mysql> grant replication slave on *.* to repluser@"%" identified by "123456"
07. Query OK, 0 rows affected, 1 warning (0.04 sec)
08. mysql>

```

3) 查看日志信息

```

01. [root@mysql11 ~]# mysql -uroot -p123qqq...A
02. mysql> show master status;
03. +-----+-----+-----+-----+-----+
04. | File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
05. +-----+-----+-----+-----+-----+
06. | master11.000001 |      441 |              |                  |                  |
07. +-----+-----+-----+-----+-----+
08. 1 row in set (0.00 sec)
09.
10. mysql>

```

步骤二：配置从服务器

1) 指定server_id

```

01. [root@mysql22 ~]# vim /etc/my.cnf
02. [mysqld]
03. server_id=22
04. :wq

```

[Top](#)

```
05. [root@mysql22 ~]# systemctl restart mysqld
```

2) 指定主服务器信息

```
01. [root@mysql22 ~]# mysql -uroot -p123qqq...A
02. mysql: [Warning] Using a password on the command line interface can be insecure
03. Welcome to the MySQL monitor.  Commands end with ; or \g.
04. Your MySQL connection id is 3
05. Server version: 5.7.17 MySQL Community Server (GPL)
06. Mysql>
07.
08. mysql> change master to master_host="192.168.4.11",master_user="repluser",
09.     -> master_password="123qqq...A",master_log_file="master11.000001",master
10. Query OK, 0 rows affected, 2 warnings (0.41 sec)
11.
12. mysql>
```

3) 启动slave进程

```
01. mysql> start slave ;
02. Query OK, 0 rows affected (0.02 sec)
03.
04. mysql>
```

4) 查看状态信息

```
01. [root@mysql22 ~]# mysql -uroot -p123qqq...A -e "show slave status\G" |grep -
02. mysql: [Warning] Using a password on the command line interface can be insecure
03. Slave_IO_Running: Yes
04. Slave_SQL_Running: Yes
05. [root@mysql22 ~]#
06.
07. [root@mysql22 ~]# mysql -uroot -p123qqq...A -e "show slave status\G" |grep -
08. mysql: [Warning] Using a password on the command line interface can be insecure
09. Master_Host: 192.168.4.11
10. [root@mysql22 ~]#
```

[Top](#)

4 案例4：配置读写分离服务

4.1 问题

配置步骤如下：

1. 安装软件
2. 修改配置文件
3. 配置数据库服务器
4. 启动服务
5. 查看服务状态
6. 查看监控信息

4.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置读写分离服务

1) 安装软件

```
01. [root@maxscale77 ~]# rpm -ivh maxscale-2.1.2-1.rhel.7.x86_64.rpm //安装软件
02. 警告: maxscale-2.1.2-1.rhel.7.x86_64.rpm: 头V4 RSA/SHA1 Signature, 密钥 ID
03. 准备中... ##### [100%]
04. 正在升级/安装...
05. 1:maxscale-2.1.2-1 (
06. [root@maxscale77 ~]#
07. [root@maxscale77 ~]# ls /etc/maxscale.cnf //主配置文件
08. /etc/maxscale.cnf
09. [root@maxscale77 ~]# ls /var/log/maxscale/ //日志目录
```

2) 修改配置文件

```
01. [root@maxscale77 ~]# vim /etc/maxscale.cnf
02.
03. [maxscale] //服务线程数量
04. threads=auto
05.
06. [server1] //第1台数据库服务器
07. type=server
08. address=192.168.4.11
09. port=3306
10. protocol=MySQLBackend
11.
12. [server2] //第2台数据库服务器
```

[Top](#)


```
13.  type=server
14.  address=192. 168. 4. 22
15.  port=3306
16.  protocol=MySQLBackend
17.
18.  [MySQL Monitor] //指定监控数据库服务器server1 和 server2
19.  type=monitor
20.  module=mysqlmon
21.  servers=server1, server2
22.  user=maxscalemon //监控用户
23.  passwd=123qqq...A //密码
24.  monitor_interval=10000
25.
26.  [Read-Write Service] //定义读写分离服务
27.  type=service
28.  router=readwritesplit
29.  servers=server1, server2
30.  user=maxscalerouter //路由用户
31.  passwd=123qqq...A //密码
32.  max_slave_connections=100%
33.
34.  [MaxAdmin Service] //定义管理服务
35.  type=service
36.  router=cli
37.
38.  [Read-Write Listener] //定义读写分离服务端口
39.  type=listener
40.  service=Read-Write Service
41.  protocol=MySQLClient
42.  port=4006 //端口号
43.
44.  [MaxAdmin Listener] //定义管理服务端口
45.  type=listener
46.  service=MaxAdmin Service
47.  protocol=maxscaled
48.  socket=default
49.  port=4016 //端口号
50.  :wq
51.  [root@maxscale77 ~]#
52.
```

[Top](#)

步骤二：配置数据库服务器

1) 创建用户：在主服务器上添加，从服务器查看即可

```

01. [root@mysql11 ~]# mysql -uroot -p123qqq...A
02. mysql> grant replication slave, replication client on *.*
03.     -> to maxscalemon@"%" identified by "123qqq...A";
04. Query OK, 0 rows affected, 1 warning (0.03 sec)
05.
06. mysql> grant select on mysql.* to
07.     -> maxscalerouter@"%" identified by "123qqq...A";
08. Query OK, 0 rows affected, 1 warning (0.02 sec)
09.
10. mysql>

```

2) 从服务器查看授权用户

```

01. [root@mysql122 ~]# mysql -uroot -p123qqq...A -e 'select user,host from mysql.'
02. mysql: [Warning] Using a password on the command line interface can be insecure
03. +-----+-----+
04. | user          | host |
05. +-----+-----+
06. | maxscalemon   | %    |
07. | maxscalerouter | %    |
08. +-----+-----+
09. [root@mysql122 ~]#

```

步骤三：启动maxscale服务

1) 启动服务

```

01. [root@maxscale77 ~]#
02. [root@maxscale77 ~]# maxscale -f /etc/maxscale.cnf
03. [root@maxscale77 ~]#

```

2) 查看服务信息（进程 和 端口）

```

01. [root@maxscale77 ~]# ps -C maxscale

```

[Top](#)

```

02.      PID TTY          TIME CMD
03.  23254 ?           00:00:00 maxscale
04.  [root@maxscale77 ~]#
05.
06.  [root@maxscale77 ~]# netstat -utnlp | grep maxscale
07.  tcp6          0      0 :::4006             :::*                  LISTEN
08.  tcp6          0      0 :::4016             :::*                  LISTEN
09.  [root@maxscale77 ~]#

```

3) 在maxscale服务本机访问管理服务，查看监控信息

```

01.  [root@maxscale77 ~]# maxadmin -uadmin -pmariadb -P4016
02.  MaxScale> list servers
03.  Servers.
04.
05.  Server          | Address          | Port | Connections | Status
06.  -----+-----+-----+-----+-----
07.  server1         | 192.168.4.11    | 3306 | 0           | Master, Running
08.  server2         | 192.168.4.22    | 3306 | 0           | Slave, Running
09.  -----+-----+-----+-----+-----
10.  MaxScale>
11.  MaxScale> exit
12.  [root@maxscale77 ~]#

```

5 案例5：准备NFS服务存储磁盘

5.1 问题

具体配置如下：

1. 添加磁盘
2. 磁盘分区
3. 创建LV
4. 格式化

5.2 方案

给NFS服务器主机添加1块10G的磁盘。如图-2所示。

[Top](#)

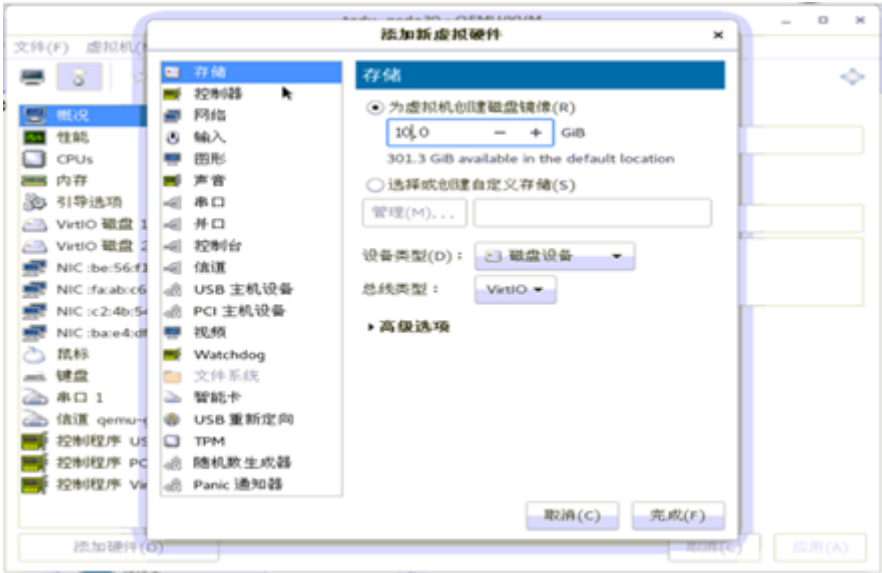


图-2

5.3

5.4 步骤

实现此案例需要按照如下步骤进行。

步骤一：磁盘分区

1) 创建分区，分1个区即可

```
01. [root@nfs30 ~]# fdisk -l /dev/vdb //查看磁盘信息
02. 磁盘 /dev/vdb: 10.7 GB, 10737418240 字节, 20971520 个扇区
03. Units = 扇区 of 1 * 512 = 512 bytes
04. 扇区大小(逻辑/物理): 512 字节 / 512 字节
05. I/O 大小(最小/最佳): 512 字节 / 512 字节
06.
07. [root@nfs30 ~]#
08. [root@nfs30 ~]# fdisk /dev/vdb //磁盘分区
09. 欢迎使用 fdisk (util-linux 2.23.2)。
10.
11. 更改将停留在内存中，直到您决定将更改写入磁盘。
12. 使用写入命令前请三思。
13.
14. Device does not contain a recognized partition table
15. 使用磁盘标识符 0x67bb10cf 创建新的 DOS 磁盘标签。
16.
17. 命令(输入 m 获取帮助): n //新建分区
18. Partition type:
19.   p primary (0 primary, 0 extended, 4 free)
20.   e extended
```

[Top](#)

```

21.  Select (default p):
22.  Select (default p): p //创建主分区
23.  分区号 (1-4, 默认 1): //回车
24.  起始 扇区 (2048-20971519, 默认为 2048): //回车
25.  将使用默认值 2048
26.  Last 扇区, +扇区 or +size {K,M,G} (2048-20971519, 默认为 20971519): //回车
27.  将使用默认值 20971519
28.  分区 1 已设置为 Linux 类型, 大小设为 10 GiB
29.
30.  命令(输入 m 获取帮助): w //保存退出
31.  The partition table has been altered!
32.
33.  Calling ioctl() to re-read partition table.
34.  正在同步磁盘。
35.  [root@nfs30 ~]#

```

2) 查看分区

```

01.  [root@nfs30 ~]# fdisk -l /dev/vdb
02.
03.  磁盘 /dev/vdb: 10.7 GB, 10737418240 字节, 20971520 个扇区
04.  Units = 扇区 of 1 * 512 = 512 bytes
05.  扇区大小(逻辑/物理): 512 字节 / 512 字节
06.  I/O 大小(最小/最佳): 512 字节 / 512 字节
07.  磁盘标签类型: dos
08.  磁盘标识符: 0x67bb10cf
09.
10.      设备 Boot      Start          End      Blocks   Id  System
11.  /dev/vdb1           2048        20971519     10484736    83  Linux
12.  [root@nfs30 ~]#

```

步骤二：挂载磁盘

1) 格式化

```

01.  [root@nfs30 ~]# mkfs.xfs /dev/vdb1 //格式化
02.  meta-data=/dev/vdb1             isize=512    agcount=4, agsize=655296 blks
03.              =                   sectsz=512   attr=2, projid32bit=1
04.              =                   crc=1        finobt=0, sparse=0

```

```

05. data = bsize=4096 blocks=2621184, imaxpct=25
06. = sunit=0 swidth=0 blks
07. naming =version 2 bsize=4096 ascii-ci=0 ftype=1
08. log =internal log bsize=4096 blocks=2560, version=2
09. = sectsz=512 sunit=0 blks, lazy-count=1
10. realtime =none extsz=4096 blocks=0, rtextents=0
11. [root@nfs30 ~]#
12. [root@nfs30 ~]# blkid /dev/vdb1 //查看
13. /dev/vdb1: UUID="81740d7a-51f1-4ce1-a830-7b6517cc778e" TYPE="xfs"
14. [root@nfs30 ~]#

```

2) 配置开机挂载

```

01. [root@nfs30 ~]# vim /etc/fstab //修改配置文件
02. /dev/vdb1 /sitedir xfs defaults 0 0
03. [root@nfs30 ~]#
04.
05. [root@nfs30 ~]# mkdir /sitedir //创建挂载目录
06.
07. [root@nfs30 ~]# chmod o+w /sitedir/ //赋予写权限
08.
09. [root@nfs30 ~]# mount -a //加载文件中所有未加载的设备
10.
11. [root@nfs30 ~]# mount | grep "/sitedir" //查看加载信息
12. /dev/vdb1 on /sitedir type xfs (rw,relatime,attr2,inode64,noquota)
13. [root@nfs30 ~]#

```

6 案例6：配置NFS服务

6.1 问题

具体配置如下：

1. 安装软件
2. 修改配置文件
3. 启动服务
4. 查看共享信息

6.2 步骤

实现此案例需要按照如下步骤进行。

[Top](#)

步骤一：部署NFS服务

1) 安装软件

```

01. [root@nfs30 ~]# yum -y install nfs-utils rpcbind
02. 已安装:
03.     nfs-utils.x86_64 1:1.3.0-0.54.el7                      rpcbind.x86_64 0:0.2.0
04.
05. 作为依赖被安装:
06.     gssproxy.x86_64 0:0.7.0-17.el7                        keyutils.x86_64 0:1.5.
07.     libbasicobjects.x86_64 0:0.1.1-29.el7                  libcollection.x86_64 0
08.     libevent.x86_64 0:2.0.21-4.el7                          libini_config.x86_64 0
09.     libnfsidmap.x86_64 0:0.25-19.el7                        libpath_utils.x86_64 0
10.     libref_array.x86_64 0:0.1.5-29.el7                      libtirpc.x86_64 0:0.2.
11.     libverto-libevent.x86_64 0:0.2.5-4.el7                  quota.x86_64 1:4.01-17
12.     quota-nls.noarch 1:4.01-17.el7                          tcp_wrappers.x86_64 0:
13.
14. 完毕!
15. [root@nfs30 ~]#

```

2) 修改配置文件

```

01. [root@nfs30 ~]#
02. [root@nfs30 ~]# vim /etc/exports
03. /sitedir *(rw)
04. :wq
05. [root@nfs30 ~]# exportfs -r //加载配置
06. [root@nfs30 ~]#

```

步骤二：启动服务

1) 启动服务

```

01. [root@nfs30 ~]# systemctl start rpcbind
02. [root@nfs30 ~]#
03. [root@nfs30 ~]# systemctl start nfs
04. [root@nfs30 ~]#
05.
06. [root@nfs30 ~]# systemctl enable nfs
07. Created symlink from /etc/systemd/system/multi-user.target.wants to /usr/lib/systemd/system/rpcbind.service.
08. [root@nfs30 ~]#
09. [root@nfs30 ~]# systemctl enable rpcbind

```

```
10. [root@nfs30 ~]#
```

2) 查看共享信息

```
01. [root@nfs30 ~]# vim /etc/fstab //修改配置文件
02. [root@nfs30 ~]# showmount -e localhost
03. Export list for localhost:
04. /sitedir *
05. [root@nfs30 ~]#
```

7 案例7：配置网站服务

7.1 问题

具体配置如下：

1. 安装软件
2. 挂载共享存储
3. 启动服务
4. 查看共享信息

7.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署网站服务

1) 安装软件

```
01. [root@web33 ~]# yum -y install httpd //网站服务软件
02. 已安装:
03. httpd.x86_64 0:2.4.6-80.el7.centos
04.
05. 作为依赖被安装:
06. apr.x86_64 0:1.4.8-3.el7_4.1 apr-util.x86_64 0:1.5.2
07. httpd-tools.x86_64 0:2.4.6-80.el7.centos mailcap.noarch 0:2.1.41
08.
09. 完毕!
10. [root@web33 ~]#
11.
12. [root@web33 ~]# yum -y install nfs-utils //访问nfs服务软件
13. 已安装:
14. nfs-utils.x86_64 1:1.3.0-0.54.el7
15.
```

[Top](#)


```

16.  作为依赖被安装:
17.      gssproxy.x86_64 0:0.7.0-17.el7                                keyutils.x86_64 0:1.5.
18.      libbasicobjects.x86_64 0:0.1.1-29.el7                        libcollection.x86_64 0
19.      libevent.x86_64 0:2.0.21-4.el7                                libini_config.x86_64 0
20.      libnfsidmap.x86_64 0:0.25-19.el7                             libpath_utils.x86_64 0
21.      libref_array.x86_64 0:0.1.5-29.el7                            libtirpc.x86_64 0:0.2.
22.      libverto-libevent.x86_64 0:0.2.5-4.el7                      quota.x86_64 1:4.01-17
23.      quota-nls.noarch 1:4.01-17.el7                              rpcbind.x86_64 0:0.2.0
24.      tcp_wrappers.x86_64 0:7.6-77.el7
25.
26.  完毕!
27.  [root@web33 ~]#

```

2) 挂载共享目录 到 /var/www/html

```

01.  [root@web33 ~]# showmount -e 192.168.4.30 //查看共享
02.  Export list for 192.168.4.30:
03.  /sitedir *
04.  [root@web33 ~]#
05.  [root@web33 ~]# vim /etc/fstab //配置开机挂载
06.  192.168.4.30:/sitedir /var/www/html nfs defaults 0 0
07.  :wq
08.
09.  [root@web33 ~]# mount -a //加载所有未加载的设备

```

步骤二：启动服务

1) 启动服务

```

01.  [root@web33 ~]# systemctl start httpd
02.  [root@web33 ~]#
03.
04.  [root@web33 ~]# systemctl enable remote-fs.target //设置远程加载服务开机运行
05.
06.  [root@web33 ~]# systemctl enable httpd
07.  Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.servi
08.
09.  [root@web33 ~]#

```

[Top](#)

2) 查看挂载信息

```
01. [root@web33 ~]#  
02. [root@web33 ~]# mount | grep "/var/www/html"  
03. 192.168.4.30:/sitedir on /var/www/html type nfs4 (rw,relatime,vers=4.1,rsiz  
04. [root@web33 ~]#
```

8 案例8：测试配置

8.1 问题

具体配置如下：

1. 测试NFS服务
2. 测试MySQL服务

8.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：测试NFS服务

- 1) 在nfs服务器创建网页 test.html

```
01. [root@nfs30 ~]# echo "web test page" > /sitedir/test.html  
02. [root@nfs30 ~]#  
03. [root@nfs30 ~]# ls /sitedir/  
04. test.html  
05. [root@nfs30 ~]#
```

- 2) 在客户端192.168.4.254 分别访问网站服务

```
01. [root@room9pc17 ~]# curl http://192.168.4.33/test.html //访问网站33主机  
02. web test page  
03. [root@room9pc17 ~]#  
04. [root@room9pc17 ~]# curl http://192.168.4.44/test.html //访问网站44主机  
05. web test page  
06. [root@room9pc17 ~]#
```

步骤二：测试MySQL服务

[Top](#)

- 1) 在主数据库服务器上添加，访问数据的连接用户

```

01. [root@mysql11 ~]# mysql -uroot -p123qqq...A
02. mysql> create database gamedb; //建库
03. Query OK, 1 row affected (0.03 sec)
04.
05. mysql> grant select,insert,update,delete on gamedb.* to yaya99@"%" identi
06. Query OK, 0 rows affected, 1 warning (0.03 sec)
07.
08. mysql> create table gamedb.user(name char(15)); //建表
09. Query OK, 0 rows affected (0.31 sec)
10.
11. mysql> select * from gamedb.user;
12. Empty set (0.01 sec)
13. mysql>

```

2) 在从服务器查看是否同步数据

```

01. [root@mysql22 ~]# mysql -uroot -p123qqq...A
02. mysql> show grants for yaya99@"%"; //查看授权用户
03. +-----+
04. | Grants for yaya99@% |
05. +-----+
06. | GRANT USAGE ON *.* TO 'yaya99'@'%' |
07. | GRANT SELECT, INSERT, UPDATE, DELETE ON `gamedb`.* TO 'yaya99'@'%' |
08. +-----+
09. 2 rows in set (0.00 sec)
10. mysql> desc gamedb.user; //查看库表
11. +-----+-----+-----+-----+-----+-----+
12. | Field | Type      | Null | Key | Default | Extra |
13. +-----+-----+-----+-----+-----+-----+
14. | name  | char(15)  | YES  |     | NULL    |      |
15. +-----+-----+-----+-----+-----+-----+
16. 1 row in set (0.01 sec)
17.
18. mysql> select * from gamedb.user;
19. Empty set (0.00 sec)
20.
21. mysql>
22. mysql> insert into gamedb.user values("xdd"); //用来验证数据读写分离
23. Query OK, 1 row affected (0.03 sec)
24.

```

[Top](#)

```

25.  mysql>
26.  mysql> select * from gamedb.user;
27.  +-----+
28.  | name |
29.  +-----+
30.  | xdd |
31.  +-----+
32.  1 row in set (0.00 sec)
33.
34.  mysql>

```

3) 测试读写分离服务

在网站服务器上，连接数据读写分离服务器77

```

01.  [root@web33 ~]# yum -y install mariadb //安装提供连接命令软件
02.  已安装:
03.  mariadb.x86_64 1:5.5.56-2.el7
04.  作为依赖被安装:
05.  mariadb-libs.x86_64 1:5.5.56-2.el7
06.  完毕!
07.  [root@web33 ~]#
08.
09.  [root@web33 ~]# mysql -h192.168.4.77 -P4006 -uyaya99 -p123qqq...A //连接读写
10.  Welcome to the MariaDB monitor.  Commands end with ; or \g.
11.  Your MySQL connection id is 23258
12.  Server version: 10.0.0 2.1.2-maxscale MySQL Community Server (GPL)
13.  Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.
14.  Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
15.  MySQL [(none)]>

```

4) 查询数据

```

01.  MySQL [(none)]> select * from gamedb.user; //显示的是从服务器上的数据
02.  +-----+
03.  | name |
04.  +-----+
05.  | xdd |
06.  +-----+
07.  1 row in set (0.00 sec)

```

[Top](#)

08. MySQL [(none)]>

5) 存储数据

```

01. MySQL [(none)]> insert into gamedb.user values("pmm"); //插入记录
02. Query OK, 1 row affected (0.05 sec)
03. MySQL [(none)]> select * from gamedb.user; //查看记录
04. +-----+
05. | name |
06. +-----+
07. | xdd |
08. | pmm |
09. +-----+
10. 2 rows in set (0.00 sec)
11.
12. MySQL [(none)]>

```

6) 在主服务器本机登录查看数据

```

01. [root@mysql11 ~]# mysql -uroot -p123qqq...A -e "select * from gamedb.user"
02. mysql: [Warning] Using a password on the command line interface can be insecure
03. +-----+
04. | name |
05. +-----+
06. | pmm |
07. +-----+
08. [root@mysql11 ~]#

```

9 案例9：配置监控服务

9.1 问题

具体配置如下：

1. 部署运行环境LNMP
2. 安装Zabbix软件
3. 初始化配置
4. 修改配置文件
5. 启动服务

[Top](#)

9.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署运行环境LNMP

1) 安装LNMP环境

```
01. [root@zabbixserver ~]# yum -y install gcc pcre-devel openssl-devel
02. [root@zabbixserver ~]# tar -xf nginx-1.12.2.tar.gz
03. [root@zabbixserver ~]# cd nginx-1.12.2
04. [root@zabbixserver nginx-1.12.2]# ./configure --with-http_ssl_module
05. [root@zabbixserver nginx-1.12.2]# make && make install
06. [root@zabbixserver ~]# yum -y install php php-mysql \
07. > mariadb mariadb-devel mariadb-server
08. [root@zabbixserver ~]# yum -y install php-fpm-5.4.16-42.el7.x86_64.rpm
09. //注意，php-fpm这个软件包在lnmp_soft/目录下
```

2) 修改Nginx配置文件

配置Nginx支持PHP动态网站，因为有大量PHP脚本需要执行，因此还需要开启Nginx的各种fastcgi缓存，加速PHP脚本的执行速度。

```
01. [root@zabbixserver ~]# vim /usr/local/nginx/conf/nginx.conf
02. ... ..
03. http{
04. ... ..
05.         fastcgi_buffers 8 16k; //缓存php
06.         fastcgi_buffer_size 32k; //缓存
07.         fastcgi_connect_timeout 300; //连接PHP的
08.         fastcgi_send_timeout 300; //发送
09.         fastcgi_read_timeout 300;
10.     location ~ /\.php$ {
11.         root            html;
12.         fastcgi_pass      127.0.0.1:9000;
13.         fastcgi_index     index.php;
14.         include           fastcgi.conf;
15.     }
16. ... ..
```

[Top](#)

3) 启动服务

启动Nginx、PHP-FPM、MariaDB服务，关闭SELinux与防火墙。

```
01. [root@zabbixserver ~]# systemctl start mariadb
02. [root@zabbixserver ~]# systemctl start php-fpm
03. [root@zabbixserver ~]# ln -s /usr/local/nginx/sbin/nginx /sbin/nginx
04. [root@zabbixserver ~]# nginx
05.
06. [root@zabbixserver ~]# firewall-cmd --set-default-zone=trusted
07. [root@zabbixserver ~]# setenforce 0
```

4) 客户端测试LNMP环境

服务器创建PHP测试页面，浏览器访问页面测试网页连通性。

```
01. [root@zabbixserver ~]# cat /usr/local/nginx/html/test.php
02. <?php
03. $i=33;
04. echo $i;
05. ?>
06. [root@zabbixserver ~]# curl http://192.168.2.5/test.php
```

步骤二：安装Zabbix软件

1) 安装源码Zabbix软件

```
01. [root@zabbixserver lnmp_soft]# yum -y install net-snmp-devel \
02. > curl-devel
03. //安装相关依赖包
04. [root@zabbixserver lnmp_soft]# yum -y install \
05. > libevent-devel-2.0.21-4.el7.x86_64.rpm
06. //注意libevent-devel这个软件包在lnmp_soft目录下有提供
07. [root@zabbixserver lnmp_soft]# tar -xf zabbix-3.4.4.tar.gz
08. [root@zabbixserver lnmp_soft]# cd zabbix-3.4.4/
09. [root@zabbixserver zabbix-3.4.4]# ./configure --enable-server \
10. > --enable-proxy --enable-agent --with-mysql=/usr/bin/mysql_config \
11. > --with-net-snmp --with-libcurl
12. // --enable-server安装部署zabbix服务器端软件
13. // --enable-agent安装部署zabbix被监控端软件
14. // --enable-proxy安装部署zabbix代理相关软件
15. // --with-mysql配置mysql_config路径
16. // --with-net-snmp允许zabbix通过snmp协议监控其他设备
```

[Top](#)

```

17. // --with-libcurl安装相关curl库文件，这样zabbix就可以通过curl连接http等服务，
18. [root@zabbixserver zabbix-3.4.4]# make && make install

```

步骤三：初始化配置

1) 创建数据库

```

01. [root@zabbixserver ~]# mysql
02. mysql> create database zabbix character set utf8;
03. //创建数据库，支持中文字符集
04. mysql> grant all on zabbix.* to zabbix@'localhost' identified by 'zabbix';
05. //创建可以访问数据库的账户与密码
06. [root@zabbixserver ~]# cd zabbix-3.4.4/database/mysql/
07. [root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < schema.sql
08. [root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < images.sql
09. [root@zabbixserver mysql]# mysql -uzabbix -pzabbix zabbix < data.sql

```

2) 安装依赖

```

01. [root@zabbixserver ~]# yum -y install php-gd php-xml
02. [root@zabbixserver ~]# yum -y install php-bcmath
03. [root@zabbixserver ~]# yum -y install php-mbstring

```

3) 修改php程序运行参数

```

01. [root@zabbixserver ~]# vim /etc/php.ini
02. date.timezone = Asia/Shanghai //设置时区
03. max_execution_time = 300 //最大执行时间
04. post_max_size = 32M //POST数据大小
05. max_input_time = 300 //服务器超时时间
06. memory_limit = 128M //内存容量
07. :wq
08. [root@zabbixserver ~]# systemctl restart php-fpm

```

步骤四：修改配置文件

[Top](#)

1) 在主数据库服务器上添加，访问数据的连接用户


```

01. [root@zabbixserver ~]# vim /usr/local/etc/zabbix_server.conf
02. DBHost=localhost //数据库主机，默认该行被注释
03. DBName=zabbix //设置数据库名称
04. DBUser=zabbix //设置数据库账户
05. DBPassword=zabbix //设置数据库密码，默认该行被注释
06. LogFile=/tmp/zabbix_server.log //设置日志
07.
08. :wq

```

步骤五：启动服务

1) 启动服务

```

01. [root@zabbixserver ~]# useradd -s /sbin/nologin zabbix //不创建用户无法启动
02. [root@zabbixserver ~]# zabbix_server //

```

2) 查看服务状态信息

```

01. [root@zabbixserver ~]# ss -ntulp |grep zabbix_server //确认连接状态，端口
02.
03. tcp LISTEN 0 128 *:10051 *: * users: (("zabbix_server",pid=23275,fd=4), ("zabb

```

10 案例10：具体配置如下(以网站服务器192.168.4.33为例)：

10.1 问题

具体配置如下：

1. 安装Zabbix软件
2. 修改配置文件(将agentd服务设置为主动模式)
3. 启动服务
4. 查看服务状态

10.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装软件

1) 安装依赖软件

```

01. [root@zabbixserver ~]# yum -y install gcc pcre-devel

```

[Top](#)

2) 安装源码zabbix软件

```
01. ]#tar -zxvf zabbix-3.4.4.tar.gz
02. ]#cd zabbix-3.4.4/
03. ]#./configure --enable-agent
04. ]# make install
```

3) 修改配置文件

```
01. ]#vim /usr/local/etc/zabbix_agentd.conf
02. #Server=127.0.0.1
03. StartAgents=0
04. ServerActive=192.168.4.55
05. Hostname=web33
06. :wq
```

步骤二：启动服务

1) 启动服务，只有端口没有进程

```
01. ]# useradd zabbix
02. ]# zabbix_agentd
03. ]# netstat -utnlp | grep :10050 没有端口
04. ]# ps -C zabbix_agentd 有进程
```

11 案例11：监控配置

11.1 问题

在监控服务器上做如下配置：

1. 创建主动模式监控模板
2. 配置自动发现
3. 查看监控信息

11.2 步骤

实现此案例需要按照如下步骤进行。

[Top](#)

步骤一：创建主动模式监控模板

- 1) 克隆监控模板，新模板名称为：Template OS Linux ServerActive。如图-3所示



图-3

2) 修改模板中的监控项目的监控模式为主动模式，如图-4所示



3) 禁用不支持主动模式的监控项目如图-5所示。

触发器	键值	间隔	历史记录	趋势	类型	应用集	状态
触发器 1	agent.version	1h	1w		Zabbix 客户端	Zabbix agent	停用的
触发器 1	agent.hostname	1h	1w		Zabbix 客户端	Zabbix agent	停用的
触发器 1	agent.ping	1m	1w	365d	Zabbix 客户端	Zabbix agent	停用的
触发器 1	kernel.maxproc	1h	1w	365d	Zabbix客户端(主动式)	OS	已启用

图-5

步骤二：配置自动发现

1) 创建自动发现规则

通过Configuration (配置) --> Discovery (自动发现) --> Create discovery rule (创建发现规则)，如图-6所示。



图-6

2) 填写规则

填写自动发现的IP范围（逗号隔开可以写多个），多久做一次自动发现（默认为1小时，仅实验修改为1m），如图-7所示。配置检查的方式：Ping、HTTP、FTP、Agent的自定义key等检查，如图-8所示。

[Top](#)

自动发现规则

名称	Web_discovery_rule
由agent代理程序自动发现	没有agent代理程序
IP范围	192.168.2.0-254
更新间隔	1m

图-7

更新间隔	1m
检查	ICMP ping
新的	
检查类型	ICMP ping
添加	取消

图-8

3) 创建Action动作

通过Configuration (配置) --> Actions Event source(事件源): 自动发现(Discovery)-->Create action (创建动作), 如图-9所示。

ZABBIX

监测中 资产记录 报表 **配置** 管理

主机群组 模板 主机 维护 **动作** 关联项事件 自动发现

动作

事件源	自动发现	创建动作
-----	------	------

图-9

4) 配置Action动作具体行为

配置动作, 添加动作名称, 添加触发动作的条件, 如图-10所示。

[Top](#)

名称 add_discovery_web

条件 标签 名称
A 主机IP地址 = 192.168.2.1-254

新的触发条件
主机IP地址 = 192.168.2.1-254
添加

已启用 ☒

更新 克隆 删除 取消

图-10

点击操作（触发动作后要执行的操作指令），操作细节：添加主机到组，与模板链接（HTT如图-11所示）。

默认接收人 Discovery: (DISCOVERY.DEVICE.STATUS) (DISCOVER

默认信息
Discovery rule: (DISCOVERY.RULE.NAME)
Device IP: (DISCOVERY.DEVICE.IPADDRESS)
Device DNS: (DISCOVERY.DEVICE.DNS)
Device status: (DISCOVERY.DEVICE.STATUS)
Device uptime: (DISCOVERY.DEVICE.uptime)
Device service name: (DISCOVERY.SERVICE.NAME)

操作 细节
添加到主机群组: Linux servers
链接到模板: Template App HTTP Service

操作细节
操作类型 添加到主机群组
主机群组 Linux servers
在此输入搜索

添加 取消

更新 克隆 删除 取消

图-11

步骤三：查看监控信息

1) 验证监控效果，如图-12所示

ZABBIX 监测中 资产记录 报表 配置 管理

仪表盘 问题 概述 Web监测 最新数据 触发器 图形 聚合图形

群组 Linux servers 主机 zabbix_client_web2 图形 CPU load

图-12

[Top](#)